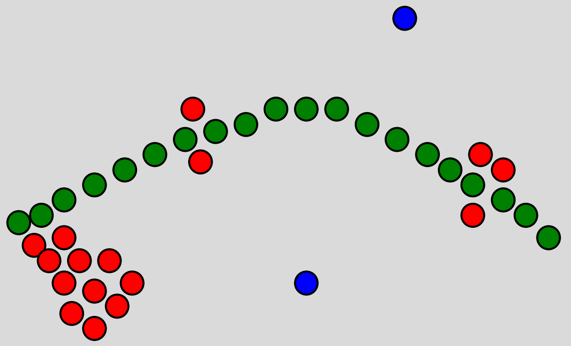
CCLRC
Rutherford Appleton Laboratory



NEAR Detector Event Reconstruction

Event Slicing

Progress report on AltReco: the Neural Net - based event reconstruction

Costas Andreopoulos *<C.V.Andreopoulos@rl.ac.uk>*

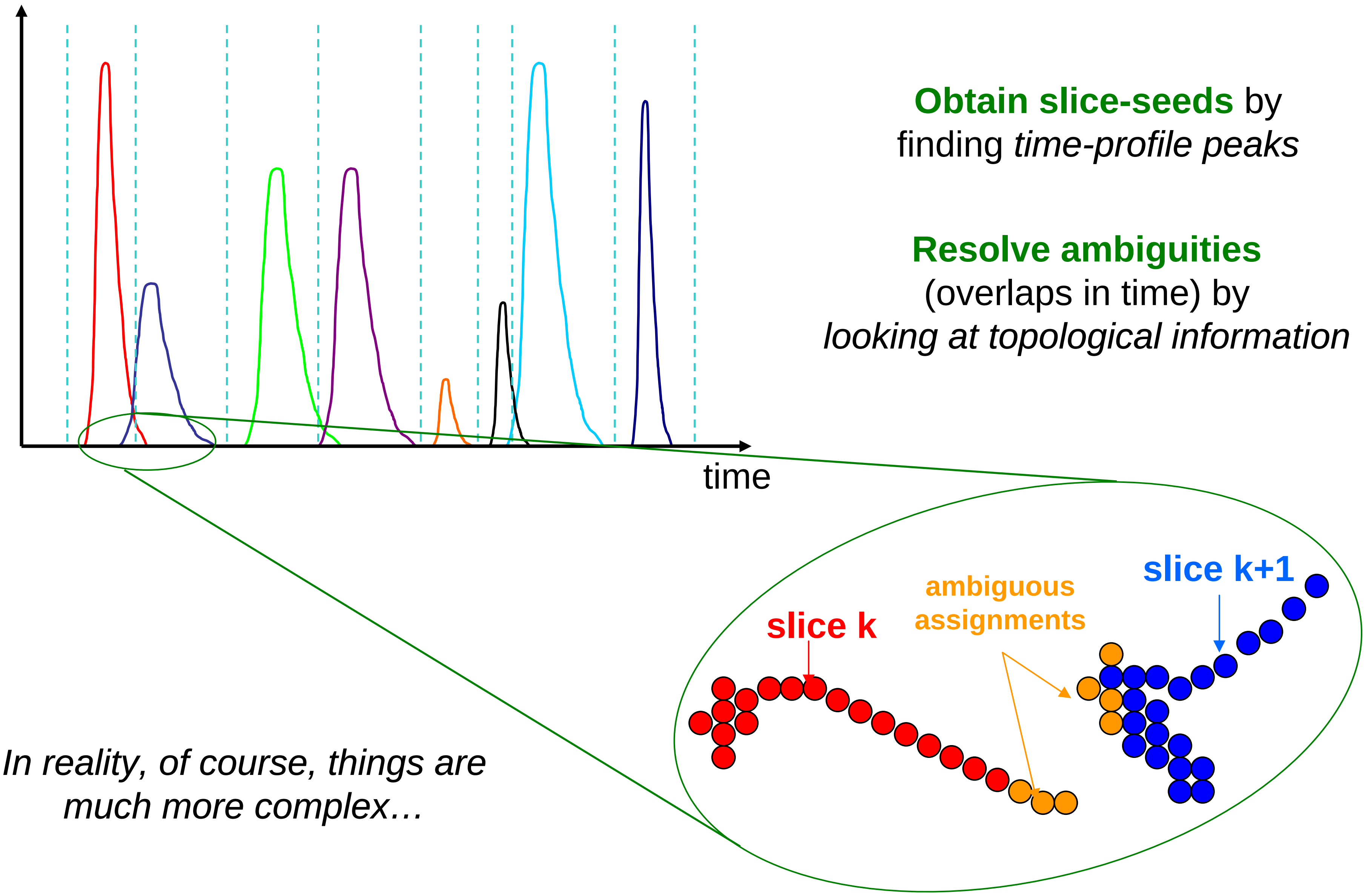
-- Near Detector Physics & Beam Systematics Working Group meeting – Oct. 07, 2003

-- Reconstruction Working Group meeting – Oct. 09, 2003

Outline

- In the collaboration meeting, I presented a fast **Neural Net**–based method for **track/shower pattern recognition** early in the reconstruction stage…
- I also flashed a page on my **event slice reconstruction** for which I was not ready to talk.
- After the collaboration meeting, I worked mostly on event slicing.
- ***In this talk:***
    - ***Event Slicing: Illustrating the basic idea***
    - ***The time-profile peak finder***
    - ***The effect of ND time resolution***
    - ***Tuning the peak finder***
    - ***A 'recursive' approach for the peak finder***
    - ***Construction of slice seeds…***
    - ***Going from "Slice-seeds" to "Slices"***
    - ***Slice refinement: 3-D clustering***
    - ***Limitations… and how to overcome them***
    - ***Adding ND muon-spectrometer hits***
    - ***Current status x2***
    - ***Future work***

Illustrating the basic idea



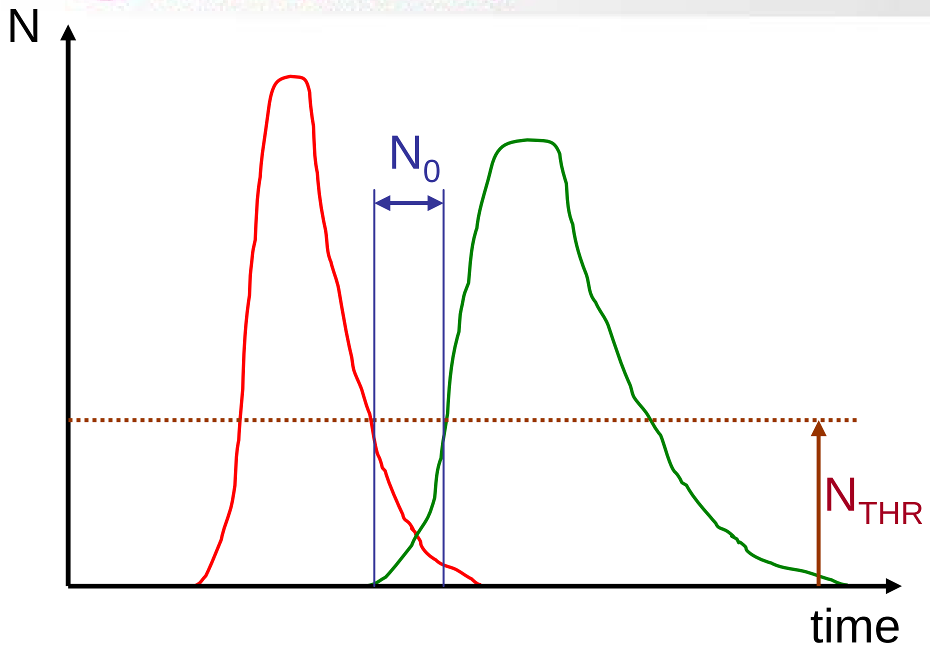
Obtain slice-seeds by finding *time-profile peaks*

Resolve ambiguities (overlaps in time) by *looking at topological information*

In reality, of course, things are much more complex…

Costas Andreopoulos <*C.V.Andreopoulos@rl.ac.uk*>

The time-profile peak finder



The peak finder is used to **identify *slice-seeds*** **not the final slices**.

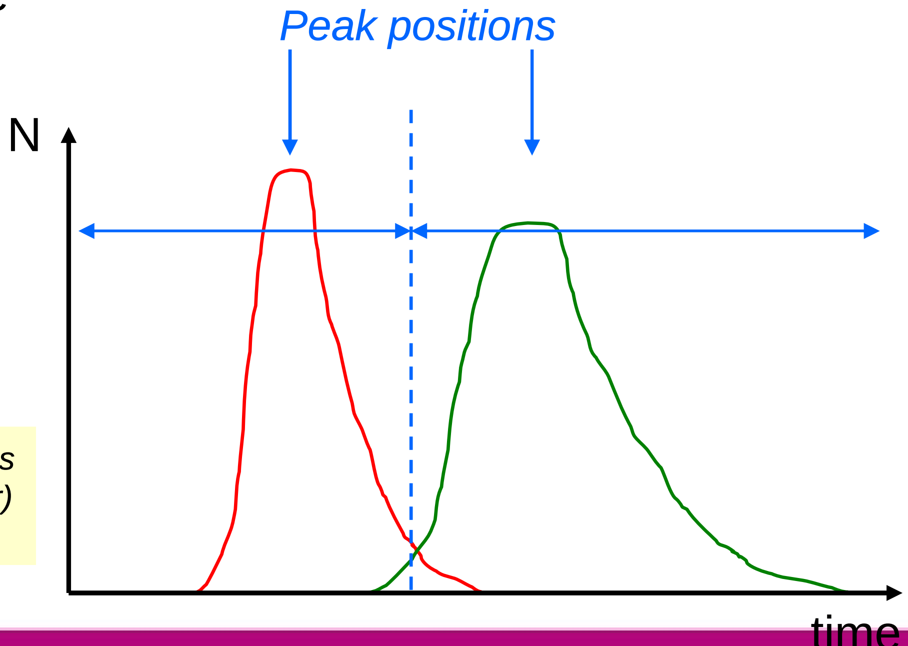
Therefore, it has to be **fairly simple** & **very fast.**

- Set a threshold N_{THR}
- Keep adding time bins to a slice-seed until you find N_0 empty time bins, after at least one non-empty time bin has been found

("empty" = content < N_{THR}, "non-empty" = content >= N_{THR})

and then

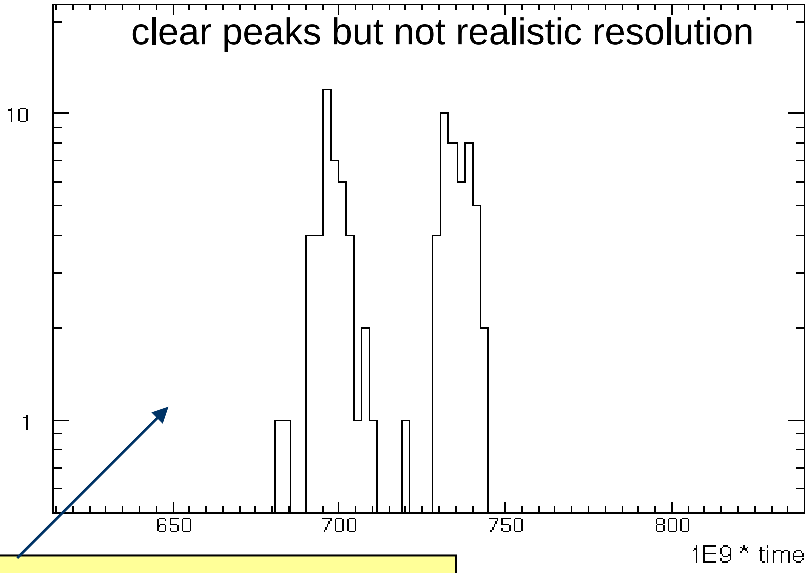
- Calculate peak position as: *<t>=Sum{qi*ti}/Sum{qi}*
- Share the 'time'-space between the two peaks proportionally to the peak charge



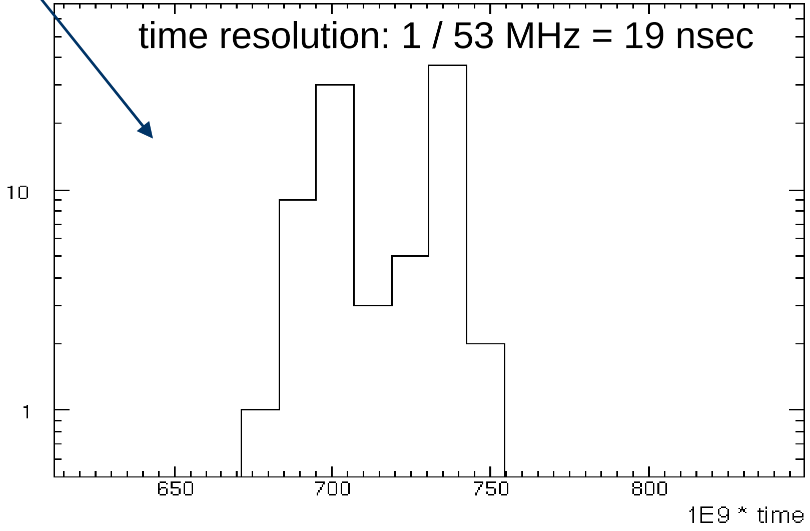
The peak-finder works with 3 externally supplied sets of params (PeakFinderConf_t = kDefault, kLowActivity, kMuSpectrometer) that are toggled internally

Costas Andreopoulos <*C.V.Andreopoulos@rl.ac.uk*>

The effect of ND time resolution

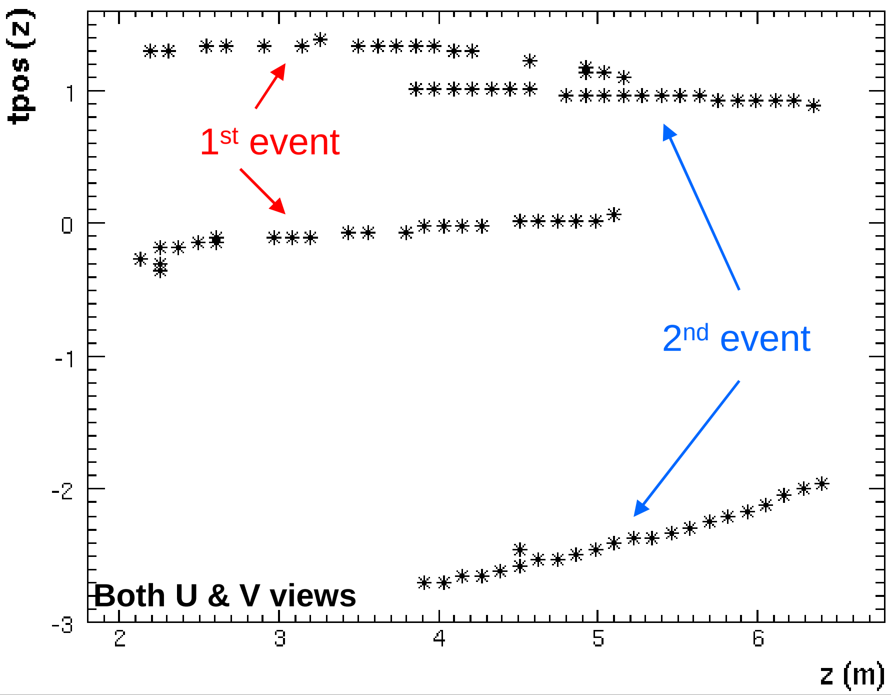


same time profile
with different time resolution



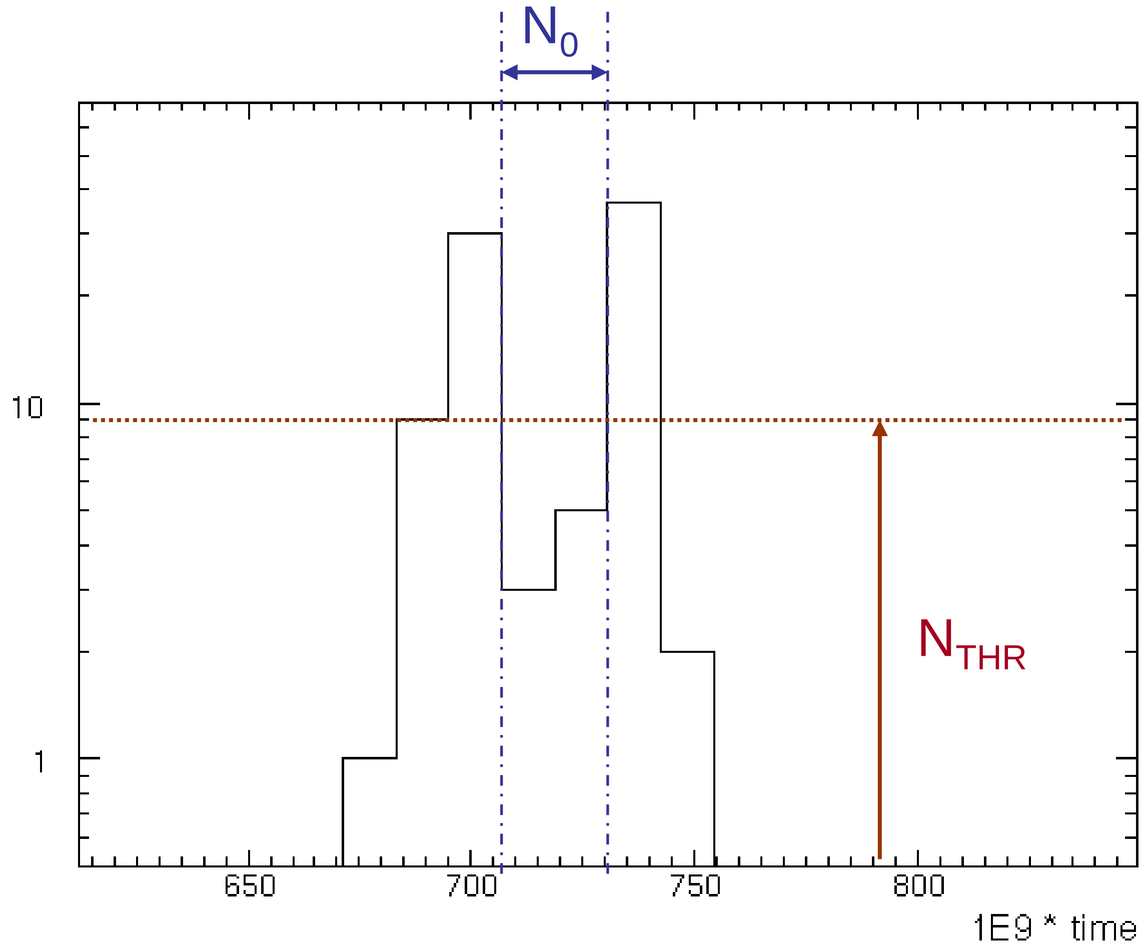
By increasing the time resolution I could, in principle, find almost all correct MC slices by the time I search for seeds !
BUT:
The ND time resolution depends on the Main Injector RF which clocks the QIE electronics

A peak finder with low threshold, for example, is easy to put both events in the same slice seed…



These events can be split, topologically, later but the seed finder must avoid it as much as possible

Tuning the peak finder



To make the peak finder more sensitive to partially overlapped events:

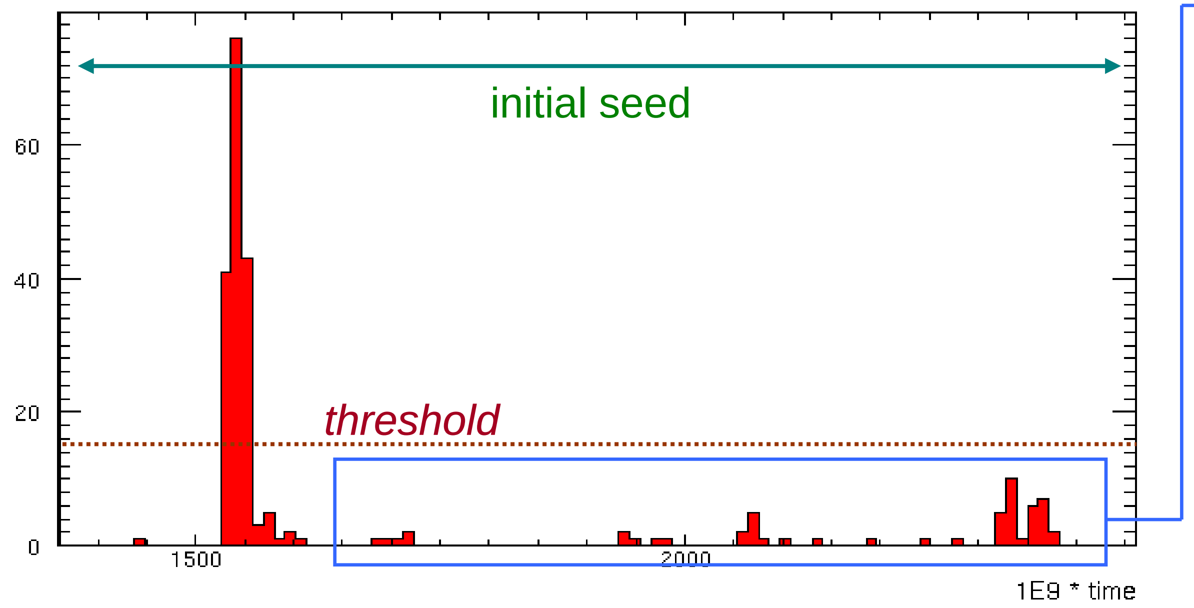
N_{THR} *is set high*

N_0 *is set low*

In this way, reasonably overlapped events are split even during slice-seed search
but this causes other complications (next >>>)

Costas Andreopoulos <*C.V.Andreopoulos@rl.ac.uk*>

A ‘recursive’ peak finder



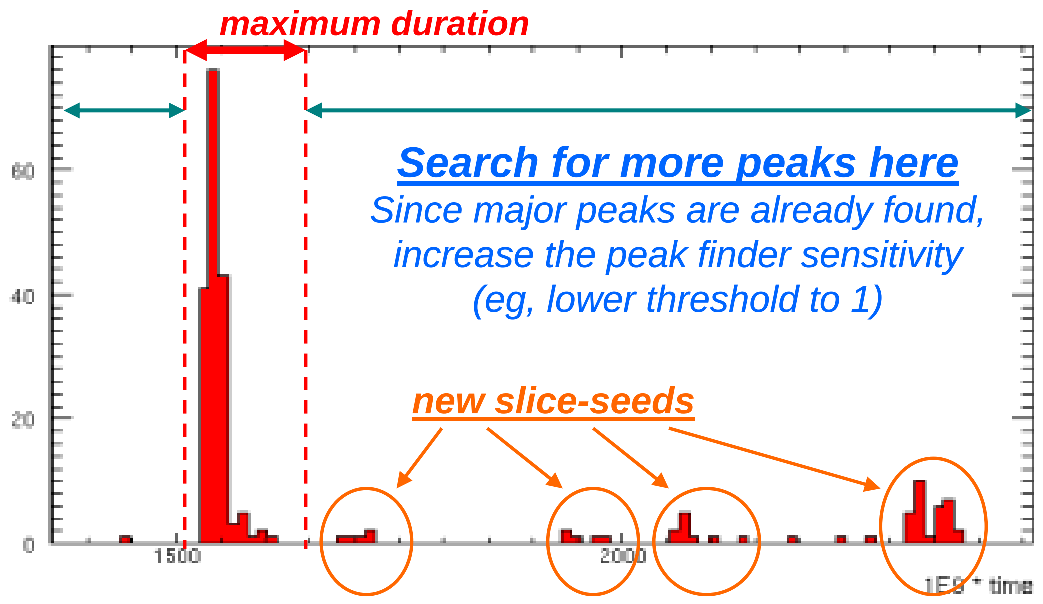
By setting the threshold high
- *NC’s,*
- *nu_e’s,*
- *low Q^2 nu_mu CC’s*
- *noise hits*

Will tend to be “attached” to a nearby slice-seed corresponding to a highly energetic event…

Following Milind’s suggestion I added in the config. the option to use a weighting scheme (eg. with charge). Weighting should reduce this effect.

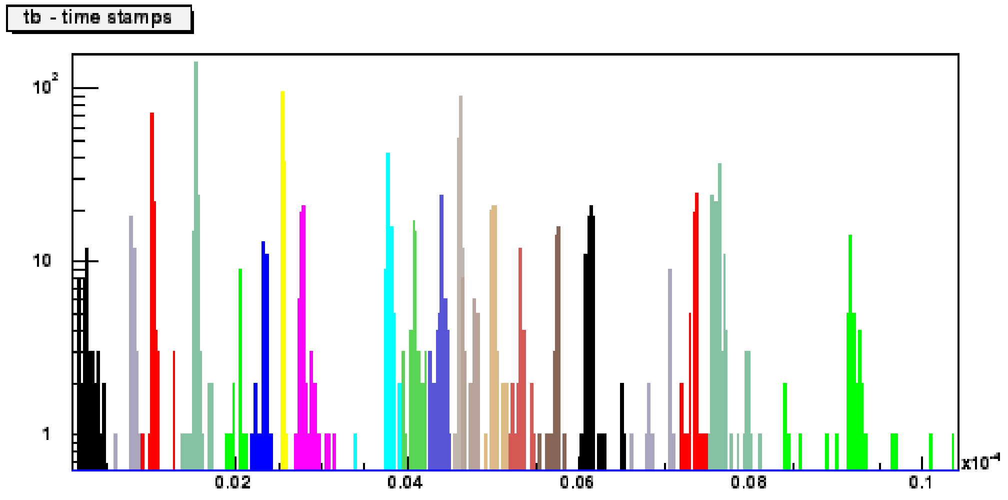
Solution:

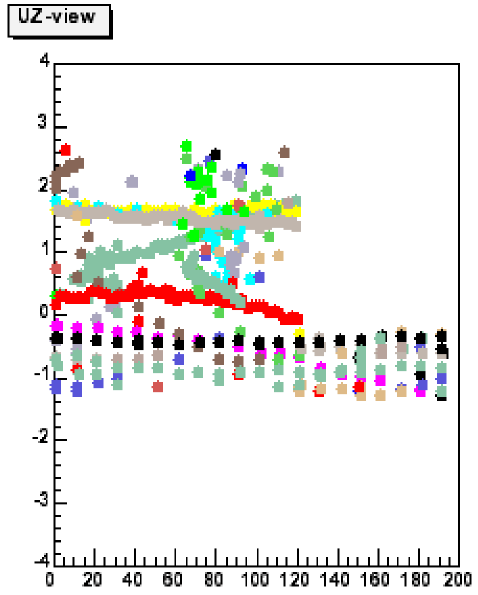
- the peak-finder **restricts the duration of any slice-seed** to a given value
- it **toggles itself** into a **configuration state** with **higher sensitivity to smaller peaks**
- then, it **<u>runs itself</u>** on the **’de-allocated’ parts of the time-profile** to find more slice seeds

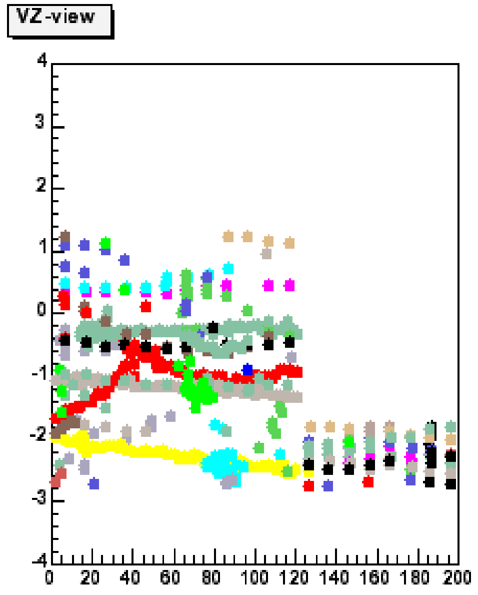


Costas Andreopoulos <*C.V.Andreopoulos@rl.ac.uk*>

So, slice seeds can be found…





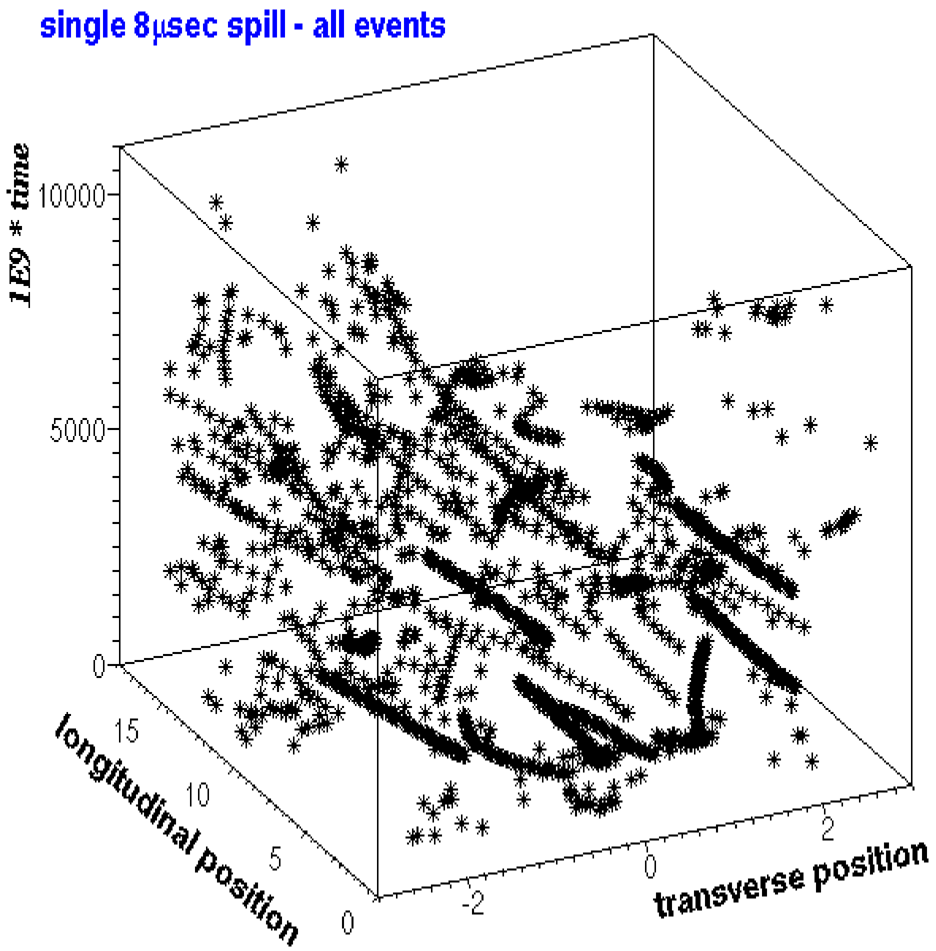


…and to 1st order they look correct and not very different from the "true" slices…

<u>'Evil things' usually live in higher orders:</u>

- *strips close to time boundaries might be assigned to a wrong slice-seed*
- *a slice-seed might contain two or more events, and therefore it must be searched for fine-structure and be split*
- *a real event might be split between one or more slice seeds and therefore these seeds need to be identified and merged*

Going from “Slice-seeds” to “Slices”



• A **slice-seed** is a *[tmin, tmax]* time interval…
• The slice-seeds are mutually exclusive.

• On the other hand, a slice is a collection of hit strips that do not correspond to “hard” time limits

• To go from slice-seeds to slices one has to:

use topological information and try to separate, in (tpos-z-time) 3-D space, events that overlap in time…

Next topic:

Slice refinement >>>

Costas Andreopoulos <*C.V.Andreopoulos@rl.ac.uk*>

CCLRC Rutherford Appleton Laboratory

Slice refinement: 3-D clustering

I have 2 options for the 3-D clustering *(both are well known & efficient)*

A "Hierarchical method"

(Minimal Spanning Trees)

Better suits the task of finding substructure within existing slices and splitting them.

work in progress…

• I need better (the new) MC before I am able to develop the part of the package that searches for slice substructure.

A "Nonhierarchical method"

(k-Means clustering)

needs a prior estimate of the number of 'clusters'
(I have the number of slice seeds)

work done… / testing

- I need to select a **clever cost function**
- Euclidian metric might not be the ultimate choice (tracks do not easily fit in a clustering algorithm although time clustering makes things better)
- Can I use some kind of **conformal mapping** prior to clustering**?**

Costas Andreopoulos <*C.V.Andreopoulos@rl.ac.uk*>

A note on what I mean by 3-D clustering

There is an overloading of the term 'clustering' - What do I mean here?

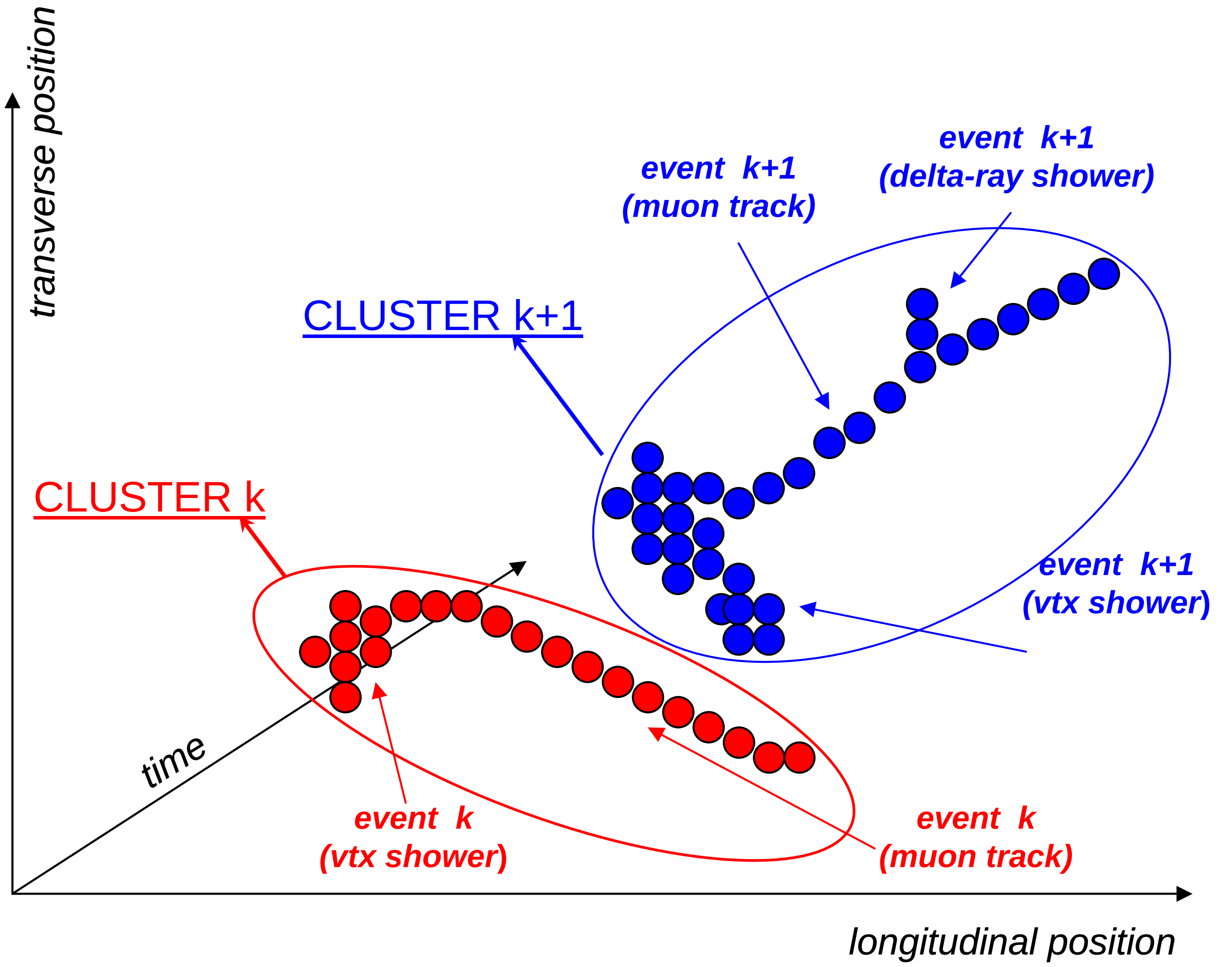
NO

~~Clustering of strips in 2 (or 3) spatial dimensions to reconstruct shower-like formations that belong to events~~

YES

Clustering of strips in a multi-dimensional space of 2 (or 3) spatial and 1 temporal dimension to reconstruct events

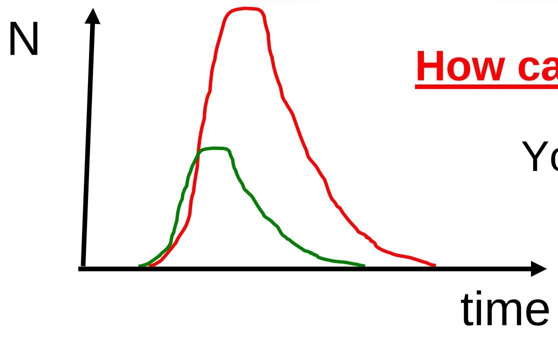
(these strips might belong to either tracks or showers)



Costas Andreopoulos <C.V.Andreopoulos@rl.ac.uk>

CCLRC Rutherford Appleton Laboratory

Limitations… and how to overcome them



How can I separate events that completely overlap in space AND time ?

You need reconstructed tracks & showers to have some chance but this happens at a later reconstruction stage…

There is a 'natural' approach:

AltReco is a Neural Net – based Reco. package after all…

- **ANNs** are used for **track / shower pattern recognition**

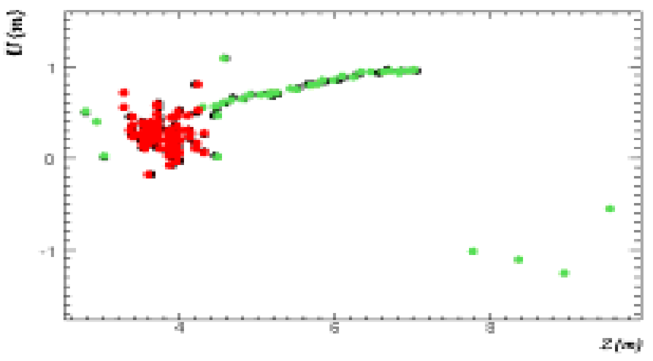
•These neural nets **do recognize neutrino 'event topology'**

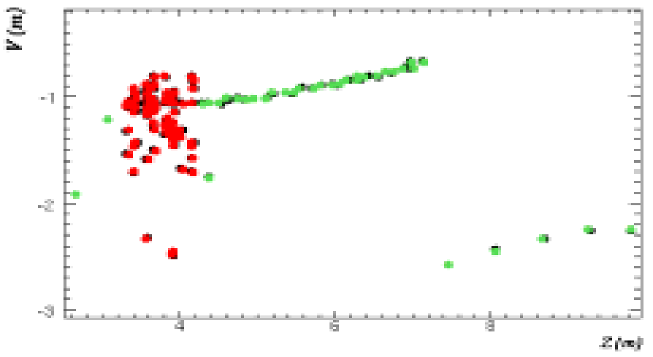
• The topology of 2 completely overlapped events should be different from the topology of a single event…

Proposal:

• Use the Neural Net to compute the likelihood that there is more than one event in a given reconstructed slice…
•Then, revisit 'event slicing' at the final reco. stages

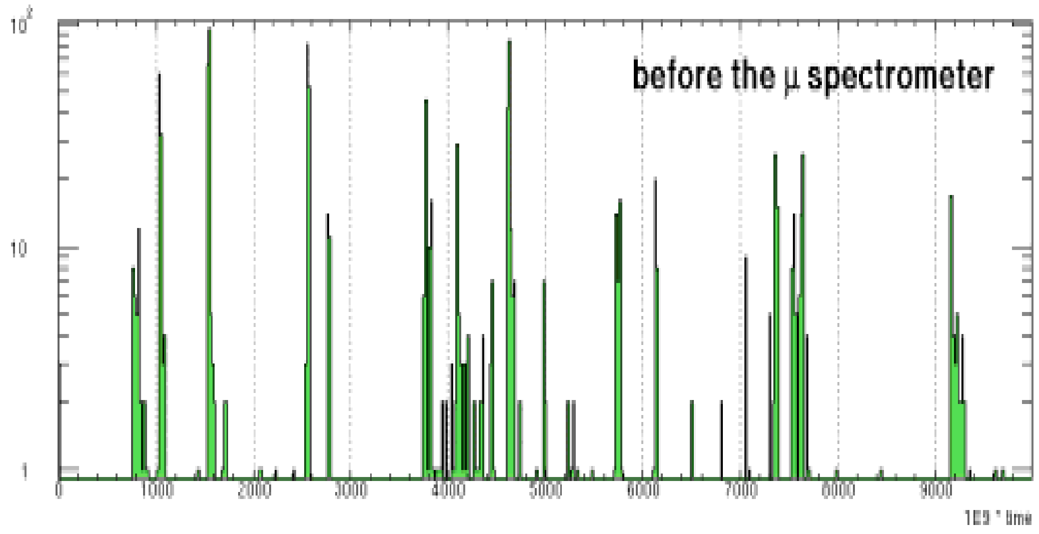
Results shown in the collab. meeting on Neural Net – based track / shower pattern recognition

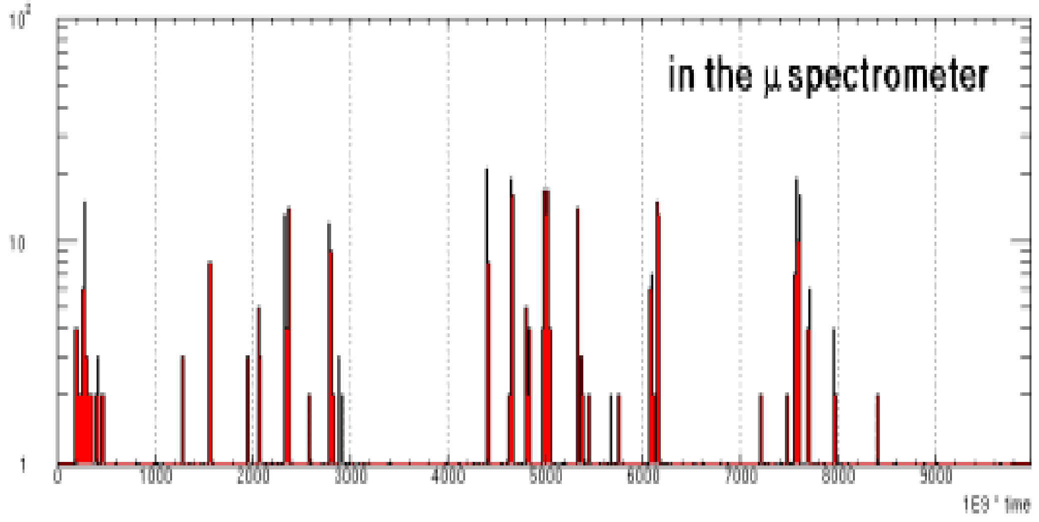




Costas Andreopoulos <*C.V.Andreopoulos@rl.ac.uk*>

Hit strips @ muon spectrometer





During the initial algorithm stages the mu-spectrometer hit strips are ignored…

They are not demuxed yet and can not be used (in a straightforward way) during the 3-D (tpos – z – time) clustering.

***After** the slices are reconstructed:*

The algorithm runs again on muon spectrometer hit strips and reconstructs more slices. Then, it tries to associate these new slices with the existing ones.

For each new slice:
If an association is found
→the slices are merged.
else
→ is added as a new slice

Hit strips @ muon spectrometer: spatial constraints

For making these slice associations, apart from temporal information, spatial information can also be used.

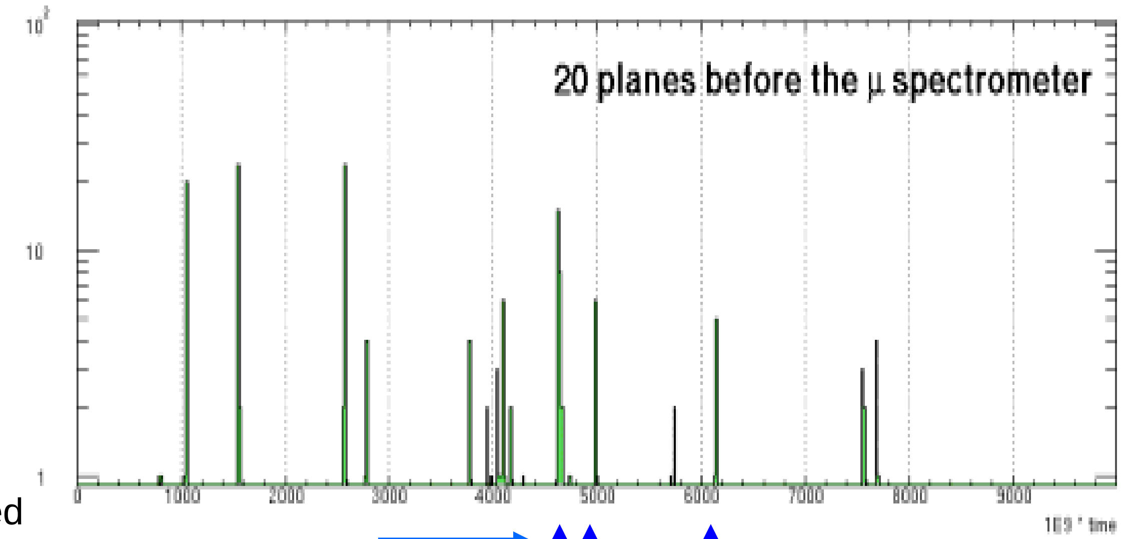
In the current approach:

A slice that has activity at the muon spectrometer might also have some activity just before the muon spectrometer.

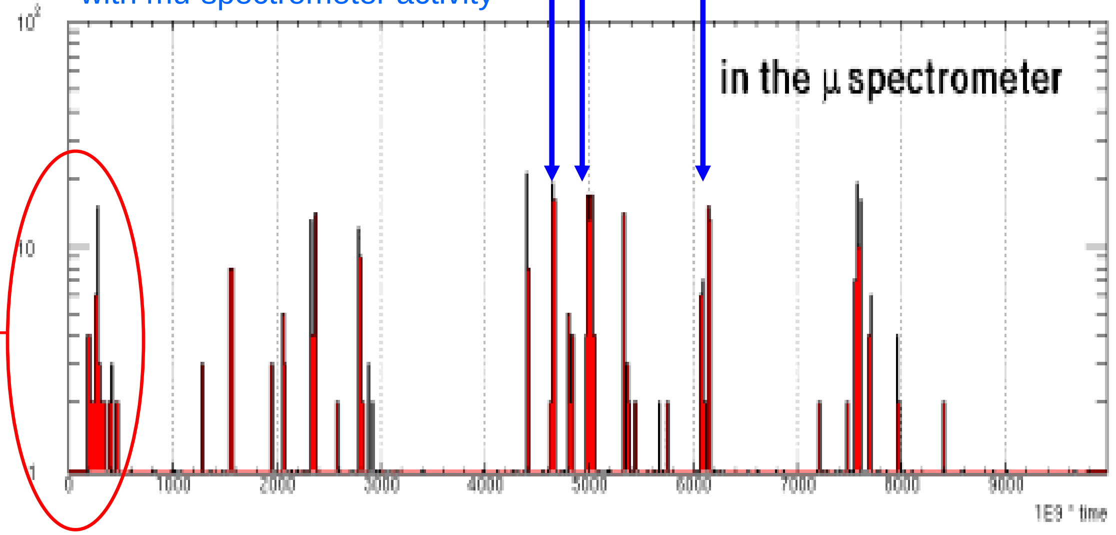
In this way some initial slices are rejected and finding associations becomes easier…

If it is needed, tpos info will be added to reduce the combinatorials

eg. nothing before the mu-spectrometer to associate this with: form a new slice to accommodate these hit strips



Examples of clear association with mu-spectrometer activity



Current Status --technical-issues

• All custom Candidate classes removed following recommendations by *George I. & Robert H.*

• ***code easier to maintain**… **removed all code duplication** in the multiple 'custom' Candidate classes.*

• *easier integration & **use of standard output tree boosted development & debugging**…*

***BUT further development** of AltReco package **critically depends on 'promised' framework's new functionality** (a Register-like 'mechanism' associated with each candidate):*

• I need to push Neural Net likelihoods to Candidate Strips (track / shower-likeness)

• I need to push Neural Net likelihoods to Candidate Slices (for overlapped events)

• The new C++ Monte Carlo is needed before I am able to go much further with the event slicing algorithm

Good news: Nathaniel T.'s work on DetSim will provide this functionality. This requires his photon transport code which will be available in < 1 month.

• Other technical problems *(extra functionality in Navigation tools, MySQL & CVS issues in my laptop's minossoft installation)* resolved *(thanks Brett V., Alex S., Nick W. & my hard-working gcc compiler…)*

This saved me lot of time.
No obstacle right now for further development / optimization of AltReco package

CCLRC Rutherford Appleton Laboratory

Future work (next 2-3 weeks)

- *test / work on recursive peak-finder methods for slice-seed construction*
- *test / work on k-Means 3-D clustering / tuning cost function parameters.*
- *add "infrastructure" for splitting slices if topology is consistent with multiple events*

cvs commit at this point

A small part of AltReco package (some older version of event slicing alg.) is committed to CVS for testing purposes…

The latest version of Event Slicing algorithm [1] will be committed later this week.

Most likely I will not commit the Neural Net parts [2] of the package yet…

[1] algorithms for building CandSliceLists & CandSlices, JobC module & macros, SRT/GNU makefile, LinkDef, wrapper classes for STL etc…

[2] Neural net functions, support classes for using Neural Nets in AltReco, algorithms for constructing CandEventLists (CandTrackLists & CandShowerLists) after applying the ANN for track / shower pattern recognition, JobC module, etc…

- More work on MST *(Minimal Spanning Tree)* – based 3-D clustering for slice splitting

DetSim / PhotonTransport at this point

- Neural nets for examining slice topology…
- … … …